

# New velocity-space discretization for continuum kinetic calculations and Fokker-Planck collisions

Matt Landreman, Darin R. Ernst

*Massachusetts Institute of Technology, Plasma Science and Fusion Center*

---

## Abstract

Numerical techniques are described for discretization of velocity space in continuum kinetic calculations. An efficient spectral collocation method is developed for the speed coordinate – the radius in velocity space – employing a novel set of non-classical orthogonal polynomials. For problems in which Fokker-Planck collisions are included, a common situation in plasma physics, a procedure is detailed to accurately and efficiently treat the field term in the collision operator. When species with disparate masses are included simultaneously, a careful extrapolation of the Rosenbluth potentials is performed. The techniques are demonstrated in neoclassical calculations of the bootstrap current and plasma flows in a tokamak.

*Keywords:* orthogonal polynomials, kinetic, Fokker-Planck, velocity space, phase space, plasma

---

## 1. Introduction

A ubiquitous situation in numerical kinetic calculations is that the distribution function must be discretized in a manner allowing both accurate integration and accurate differentiation. Integration is needed because typically moments of the distribution function such as density and velocity are of interest. Differentiation is needed due to both the collisionless terms in the kinetic equation and also for velocity-space diffusion in collisions. Spherical or cylindrical coordinates are natural for velocity space, meaning the

---

*Email address:* [landrema@mit.edu](mailto:landrema@mit.edu) (Matt Landreman)  
*Preprint submitted to J. Comp. Phys.*

*October 22, 2012*

normalized spherical or cylindrical radius coordinate  $x$  lies in the semi-infinite domain  $[0, \infty)$ . The distribution function often has a Maxwellian envelope, meaning it behaves as  $\propto \exp(-x^2)$  as  $x \rightarrow \infty$ .

Many discretization schemes for the radial coordinate are possible and many have been explored in the literature[1, 2, 3, 4, 5], each with advantages and disadvantages regarding the above requirements. A uniform grid allows modest accuracy at both integration and differentiation using finite difference/volume/element methods. To use a uniform grid,  $x$  may either be mapped to a finite interval using a coordinate transformation, or else the fact that the distribution function is exponentially small for  $x \gtrsim 6$  may be used to truncate the  $x$  grid above some  $x_{\max}$ . Alternatively, Gaussian abscissa permit extremely accurate integration but generally only low-order differentiation if finite difference/volume/element methods are applied to the unequally spaced grid. A Chebyshev grid permits both spectrally accurate differentiation and integration[6, 7], but Chebyshev grids involve a high density of nodes near the endpoints of a finite interval, and are therefore poorly suited for the semi-infinite domain of  $x$  and  $\propto \exp(-x^2)$  dependence typical of distribution functions. Methods using Laguerre or associated Laguerre polynomials have also been used, but as we will show, these methods can perform surprisingly poorly due to a nonanalytic Jacobian in the coordinate transformation from energy to speed.

In this work, we present a new approach for discretizing velocity space. The approach permits both spectrally accurate integration and differentiation on the semi-infinite domain  $[0, \infty)$  for functions with Maxwellian-like  $\propto \exp(-x^2)$  dependence for large normalized speed  $x$ . The method is collocation rather than modal in nature, i.e. the function is known on a set of grid points (abscissae) rather than being explicitly expanded in a set of modes. As such, the method is well suited for nonlinear computations in addition to linear ones.

Our method derives from a set of previously unexplored orthogonal polynomials. The semi-infinite integration domain in the orthogonality relation for these new polynomials is identical to that of the Laguerre polynomials. However, the use of a different weight  $e^{-x^2}$  in place of  $e^{-y}$  results in polynomials with superior properties for the calculations of interest.

In kinetic calculations for plasmas, it is often important to accurately include collisions

in the kinetic equation, and the Fokker-Planck operator[8] is the best available description of collisions. This operator may be written in terms of “Rosenbluth potentials”, which are defined in terms of the distribution function through a pair of Poisson equations. A complication of the Fokker-Planck operator is that the Rosenbluth potentials vary as powers of  $x$  rather than as  $\exp(-x^2)$  for large  $x$ , which means discretization schemes that work well for the distribution function may not work well for the potentials. Accurate numerical schemes for handling the Fokker-Planck operator in plasma computations have been a subject of great interest [9, 10, 4, 11]. Here we will develop an efficient approach to incorporating the Rosenbluth potentials in the kinetic equation, carefully accounting for their behavior at large  $x$  to maintain high precision even for coarse grid resolution.

The new techniques we discuss are demonstrated in several applications. First, we compute the bootstrap current in a tokamak plasma. Secondly, we calculate the flows of multiple ion species in a tokamak. Both of these computations require the solution of drift-kinetic equations, equations in which both the collision operator and other kinetic terms appear. Using the new velocity discretization described here, we find that only 4-6 grid points in  $x$  are required for the desired level of convergence. For comparison, previous numerical methods for the same problems[12, 5, 13] have typically used tens or hundreds of grid points or modes in  $x$ . Even for gyrokinetic simulations of plasma turbulence, which commonly use  $\sim 16$  energy gridpoints, this new energy grid may result in a significant speedup.

Despite the exceptional accuracy of the method reported here, finite difference and finite volume methods for differentiation do have the advantage that they can be designed to maintain conservation properties exactly[14], which can be important in time-dependent simulations. A hybrid scheme can be used, with the new scheme used to determine the grid points and integration weights, with finite differencing used to evaluate derivatives maintaining exact conservation.

## 2. Spectral collocation scheme for velocity space

For a variety of problems in kinetic theory, either with or without collisions, it is useful to represent the distribution function in either spherical or cylindrical coordinates in velocity space. The dimensional coordinates  $v$  (the spherical radius in velocity space)

or  $v_\perp$  (the cylindrical radius in velocity space) then arise. Either coordinate may be normalized for numerical work by the thermal speed  $v_{\text{th}} = \sqrt{2T/m}$  where  $T$  is a typical temperature and  $m$  is the mass of the particle species. We define  $x = v/v_{\text{th}}$  or  $x = v_\perp/v_{\text{th}}$  as appropriate.

For large  $x$ , distribution functions typically behave like a Maxwellian,  $\propto \exp(-x^2)$ . It is therefore natural to represent the distribution function as a sum of polynomials  $P_n^k$  that are orthogonal according to the relevant weight and domain:

$$\int_0^\infty P_N^k(x) P_n^k(x) x^k e^{-x^2} dx = M_n^k \delta_{N,n} \quad (1)$$

where  $k$  is any number greater than  $-1$ , and  $M_n^k$  is some normalization. Notice (1) differs from the defining orthogonality relations for both the associated Laguerre and Hermite polynomials, which are, respectively,

$$\int_0^\infty L_N^m(y) L_n^m(y) y^m e^{-y} dy = \frac{\Gamma(n+m+1)}{n!} \delta_{N,n} \quad (2)$$

(i.e. polynomials in  $x^2$  rather than  $x$ ) and

$$\int_{-\infty}^\infty H_N(x) H_n(x) e^{-x^2} dx = 2^n n! \sqrt{\pi} \delta_{N,n} \quad (3)$$

(different range of integration than (1)). Laguerre polynomials are the special case of associated Laguerre polynomials with  $m = 0$ . There are several reasons why it is preferable to use polynomials in speed  $x$  rather than polynomials in energy  $y = x^2$ , i.e why the new polynomials are preferable to Laguerre or associated Laguerre polynomials. These reasons will be developed throughout the remainder of this section. As initial motivation, consider that the new polynomials can represent both even and odd powers of  $v$  or  $v_\perp$ , whereas the associated Laguerre polynomials can represent only even powers.

In our experience, the choice  $k = 0$  tends to yield the fastest convergence for the problems we consider in the following sections, so for the rest of this paper we consider the polynomials  $P_n = P_n^0$ . It is straightforward to generalize all results and algorithms to the case of different  $k$ .

The first few polynomials may be computed iteratively using the following Gram-Schmidt procedure (though this method turns out to be poorly conditioned when  $n$  is large). The polynomial  $P_n(x)$  has  $n+1$  coefficients, which may be determined by imposing orthogonality with respect to  $P_0$  through  $P_{n-1}$  and enforcing the normalization, for a

total of  $n + 1$  constraints. The first few polynomials  $P_n$ , normalized so the leading coefficient is 1, are thus

$$\begin{aligned}
P_0(x) &= 1 \\
P_1(x) &= x - \frac{1}{\sqrt{\pi}} \\
P_2(x) &= x^2 - \frac{\sqrt{\pi}}{\pi - 2}x + \frac{4 - \pi}{2(\pi - 2)} \\
P_3(x) &= x^3 - \frac{3\pi - 8}{2\sqrt{\pi}(\pi - 3)}x^2 + \frac{10 - 3\pi}{2(\pi - 3)}x - \frac{16 - 5\pi}{4\sqrt{\pi}(\pi - 3)}.
\end{aligned} \tag{4}$$

The same polynomials normalized so  $M_n^0 = 1$  are

$$\begin{aligned}
P_0(x) &= \frac{\sqrt{2}}{\pi^{1/4}} = 1.0623 \\
P_1(x) &= 2.4921x - 1.4060 \\
P_2(x) &= 4.2656x^2 - 6.6228x + 1.6037 \\
P_3(x) &= 6.0027x^3 - 17.0392x^2 + 12.1931x - 1.7463.
\end{aligned} \tag{5}$$

The appearances of  $\sqrt{\pi}$  and  $\pi$  in (4) highlight that the  $P_n$  polynomials do not seem to be simply related to any of the classical orthogonal polynomials.

Figure 1 shows the first few normalized polynomials, multiplied by  $e^{-x^2}$ . Figure 2 shows the corresponding plot for the Laguerre polynomials. (Associated Laguerre polynomials look very similar.) These figures illustrate the first reason the new polynomials are preferable: they have structure localized more in the region where the particle density is high ( $x \lesssim 2$ ) compared to the  $L_n^m(x^2)\exp(-x^2)$  modes. For example, the new polynomial modes have significant differences from each other in the range  $x < 1$  – so they can accurately resolve structure in this region where many particles reside – whereas the Laguerre modes are very similar to each other in the same range. Even with 10 Laguerre modes, no structure can be resolved within  $x < 0.5$  since there are no nodes there, whereas the new modes permit ample resolution of this region. The same issue can be seen in figures 3 and 4, which show the grids for the corresponding spectral collocation methods. In other words, the zeros of the  $P_n(x)$  are plotted in Figure 3, and the square roots of the zeros of the  $L_n^{(m)}(y)$  are plotted in Figure 4 to show the grid points in speed rather than energy. The Laguerre and associated Laguerre bases tends to “waste” more nodes in the region  $x > 2$  where the distribution function is nearly zero.

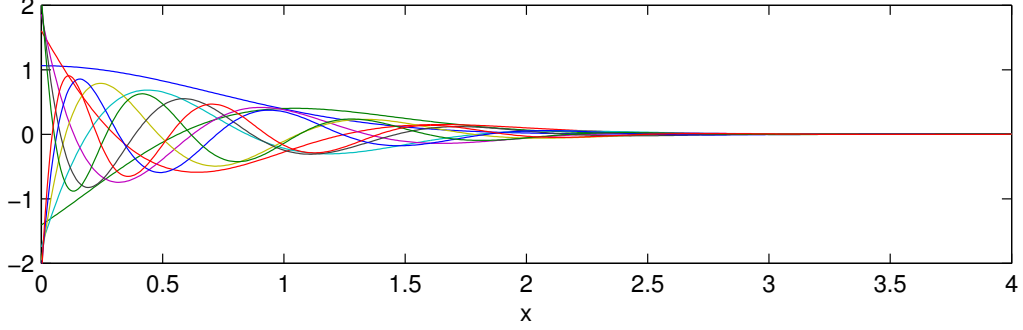


Figure 1: (Color online) The first 10 new polynomial modes  $P_n(x) \exp(-x^2)$ , normalized so  $M_n^0 = 1$ .

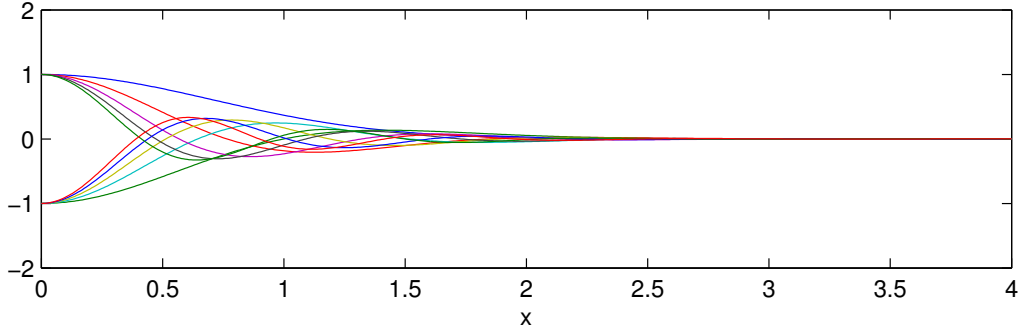


Figure 2: (Color online) The first 10 Laguerre modes  $L_n(x^2) \exp(-x^2)$  (normalized).

It should be mentioned that the abscissae can be scaled by a factor slightly less than 1 without destroying the spectral convergence properties while achieving better resolution at small values of  $x$ .

The clustering of nodes for the new polynomials near  $x = 0$  turns out to be a major advantage when multiple particle species are considered simultaneously, discussed in section 5, since high resolution is needed in this region when the species masses are disparate.

As with any set of orthogonal polynomials, the set  $P_n$  is associated with a Gaussian integration scheme, with the abscissae corresponding to the zeros of the polynomials. These abscissae and Gaussian integration weights may be computed for any orthogonality relation using the “Stieltjes procedure”, an alternative method to construct the polynomials which is better conditioned numerically than the direct orthogonalization

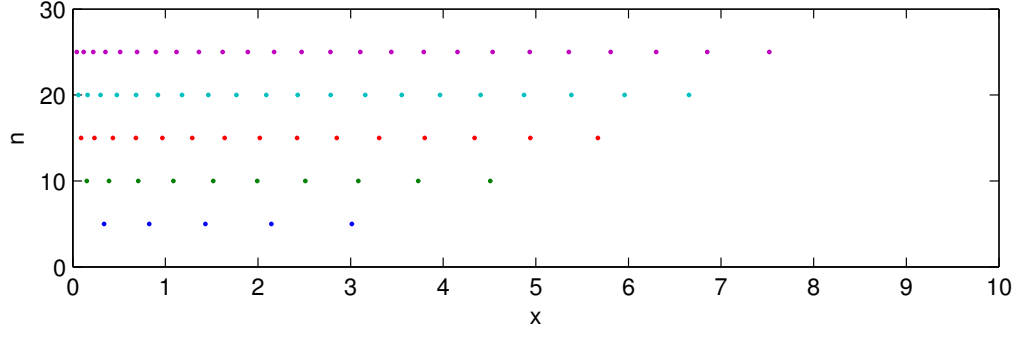


Figure 3: (Color online) Zeros of the new polynomials  $P_n(x)$ , i.e. the speed grid for Gaussian integration and for the new spectral collocation method.

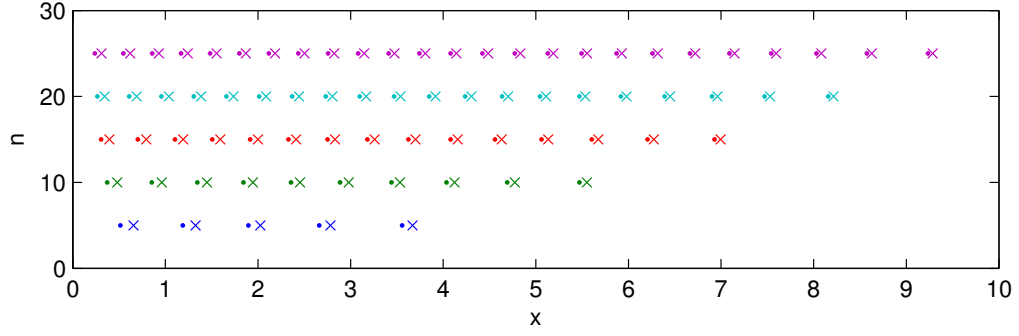


Figure 4: (Color online) Values of  $x = \sqrt{y}$  for zeros of the Laguerre polynomials  $L_n(y)$  (dots) and associated Laguerre polynomials  $L_n^{(1/2)}(y)$  (crosses), i.e. the speed grids for Gaussian integration and for spectral collocation methods.

approach described above. The theory underlying the Stieltjes procedure is detailed in Refs. [15, 16, 17], with Fortran 77 subroutines implementing the algorithms in Ref. [18]. Another practical implementation of the algorithm may be found in Ref. [19], with documentation given in section 4.6 of Ref. [20]. In brief, the algorithm involves first iteratively computing the coefficients  $a_j$  and  $b_j$  of the two-term recursion relation for the polynomials,

$$P_{j+1}(x) = (x - a_j)P_j(x) - b_j P_{j-1}(x), \quad (6)$$

which may be done by evaluating

$$a_j = \frac{\int_0^\infty P_j^2(x) x e^{-x^2} dx}{\int_0^\infty P_j^2(x) e^{-x^2} dx} \quad (7)$$

$$b_j = \frac{\int_0^\infty P_j^2(x) e^{-x^2} dx}{\int_0^\infty P_{j-1}^2(x) e^{-x^2} dx}. \quad (8)$$

The integrals in (7)-(8) are performed using adaptive quadrature, evaluating  $P_j(x)$  using the recursion relation. The resulting recursion coefficients are then used to form the tridiagonal Jacobi matrix:

$$\mathbf{J} = \begin{pmatrix} a_0 & \sqrt{b_1} & & & \\ \sqrt{b_1} & a_1 & \sqrt{b_2} & & \\ & \ddots & \ddots & \ddots & \\ & & \sqrt{b_{n-2}} & a_{n-2} & \sqrt{b_{n-1}} \\ & & & \sqrt{b_{n-1}} & a_{n-1} \end{pmatrix}. \quad (9)$$

As shown in the aforementioned references, the eigenvalues of this matrix give the abscissae. Furthermore, the first elements of the normalized eigenvectors may be squared and multiplied by  $\int_0^\infty e^{-x^2} dx = \sqrt{\pi}/2$  to obtain the integration weights. This algorithm is numerically well conditioned, and produces a grid with extremely accurate integration properties, as we will demonstrate shortly.

The above procedure produces a grid with no point at  $x = 0$ . If a point is needed there to impose a boundary condition, i.e. if a Gauss-Radau grid is desired, the bottom-right element of the Jacobi matrix may be replaced by  $-b_{n-1}P_{n-2}(0)/P_{n-1}(0)$ , as discussed in Section 6.2 of Ref. [17]. The abscissae and weights are then computed from the eigenvalues and eigenvectors as before. The resulting grid will still be extremely accurate for integration, but slightly less so than the grid with no point at  $x = 0$ .



As mentioned previously, for kinetic calculations we need not only an accurate integration scheme for the grid, but also an accurate differentiation matrix. The spectral differentiation matrix for any set of nodes and any weighting function may be computed using the *poldif* algorithm described in Refs. [21, 22, 23]. The algorithm amounts to representing a function on the grid as a weighted sum of the polynomials, differentiating the polynomials, and evaluating the result on the grid. We apply the algorithm using the weight  $e^{-x^2}$ , and using the abscissae calculated from Stieljes procedure above.

To demonstrate the efficacy of the new collocation method, figures 5-9 compare various discretization methods for both integration and differentiation of several functions. We first choose the function  $x^2 \cos(x - 1/2) \exp(-x^2)$  because 1) for large  $x$  it has a Maxwellian-like envelope, 2) for small  $x$  it has the  $x^2$  behavior typical of physical velocity moment integrands near  $x = 0$ , and 3) due to the  $\cos(x - 1/2)$  factor it cannot be exactly represented by any of the polynomial approximation methods, and its Taylor series contains both even and odd powers of  $x$ . We measure the accuracy of differentiation by comparing the numerical derivative to the analytical derivative at whichever grid point is closest to 1. For the new method, results are shown for both the standard Gauss abscissae (no point at  $x = 0$ ) and for Gauss-Radau points (including  $x = 0$ ). It can be seen that the new schemes perform substantially better in both integration and differentiation than other methods. The same results are reported in tabular form in Tables 1-2. We have performed a similar analysis to many functions besides that in figure 5.a, and the behavior seen in 5.b-c is typical.

The poor performance of both the Laguerre and associated Laguerre methods in this comparison may be understood as follows. These grids can allow very accurate integration and differentiation, but only for functions that are analytic in  $y = \sqrt{x}$ , the argument of the polynomials in the orthogonality relation that defines them. Since  $\sqrt{x}$  is nonanalytic at  $x = 0$ , the accuracy of Laguerre and associated Laguerre methods fails for general functions of  $x$ . This issue has a practical implication for kinetic computations. In kinetic codes, it is common to want both the density moment ( $\int d^3v f \propto \int_0^\infty dx x^2 f = \frac{1}{2} \int_0^\infty dy \sqrt{y} f$ ) and velocity moment ( $\int d^3v \mathbf{v} f \propto \int_0^\infty dx x^3 f = \frac{1}{2} \int_0^\infty dy y f$ ) of the distribution function  $f$ . Even if  $f$  is analytic in  $y$ , the spectral accuracy of Gaussian integration is lost when computing the density moment using a Laguerre grid due to the  $\sqrt{y}$  factor,

as noted in Refs. [2, 3]. This problem could be avoided using an associated Laguerre grid with  $m = 1/2$ , so the nonanalytic factor  $\sqrt{y}$  is incorporated into the integration weight. However, then accuracy would be lost for the velocity integrand, since it includes an additional factor of  $\sqrt{y}$ . By using polynomials in  $x$  rather than polynomials in  $y$ , good accuracy can be retained for both integrals simultaneously.

The plots also include results for a uniform grid and Chebyshev grid covering the interval  $[0, 5]$ . We choose  $x = 5$  as a reasonable cutoff since the model function is exponentially small ( $< 3 \times 10^{-10}$ ) for  $x > 5$ . For the uniform grid, integration is performed using the trapezoid rule, and differentiation is performed using 5-point finite differencing.

Also shown are results for several schemes in which  $x \in [0, \infty)$  is mapped to a new variable  $s$  in the finite interval  $[0, 1]$ . We considered the three maps  $s = -\ln(1 - x)$ ,  $x = \tan(\pi s/2)$ , and  $x = s/(1 - s)$ , applying either a uniform or Chebyshev grid to  $s$ . Results are shown for the map which produced the most accurate results. The direct Chebyshev grids in  $x$  and  $s$  perform roughly equally well to each other, while the uniform grid in  $s$  tends to outperform the uniform grid in  $x$ .

Results are also plotted for the collocation scheme used in the gyrokinetic code GS2, detailed in Ref. [2]. The same grid is also used in the astrophysics code AstroGK[3]. In this scheme, the grid consists of  $n - m$  Gauss-Legendre points on the interval  $[0, 2.5]$ , together with  $m$  Gauss-Laguerre points on the interval  $[2.5, \infty)$ , where  $m = 1$  for  $n \leq 12$  and  $m = 2$  otherwise. The philosophy behind this approach is to only use the Gauss-Laguerre points away from the singularity at  $x = 0$ . Indeed, the GS2 approach shows substantially better convergence of the integral than the Laguerre-only or associated Laguerre methods. The actual GS2 fortran subroutines are used to generate both the abscissae and weights for the figures here. Convergence of the integral stagnates around  $10^{-7} - 10^{-8}$  since no more than 2 points are ever used in the interval  $[2.5, \infty)$ ; if  $m$  is increased beyond 2, the GS2 method can reach machine precision. The derivative is computed using a 3-point finite-difference stencil, just as is used for the second derivative in GS2. This low-order scheme was used in GS2 because it allowed exact numerical conservation of particles in velocity space, which higher-order methods generally do not allow.

Figures 6-7 show interesting effects arising when the various integration and differen-

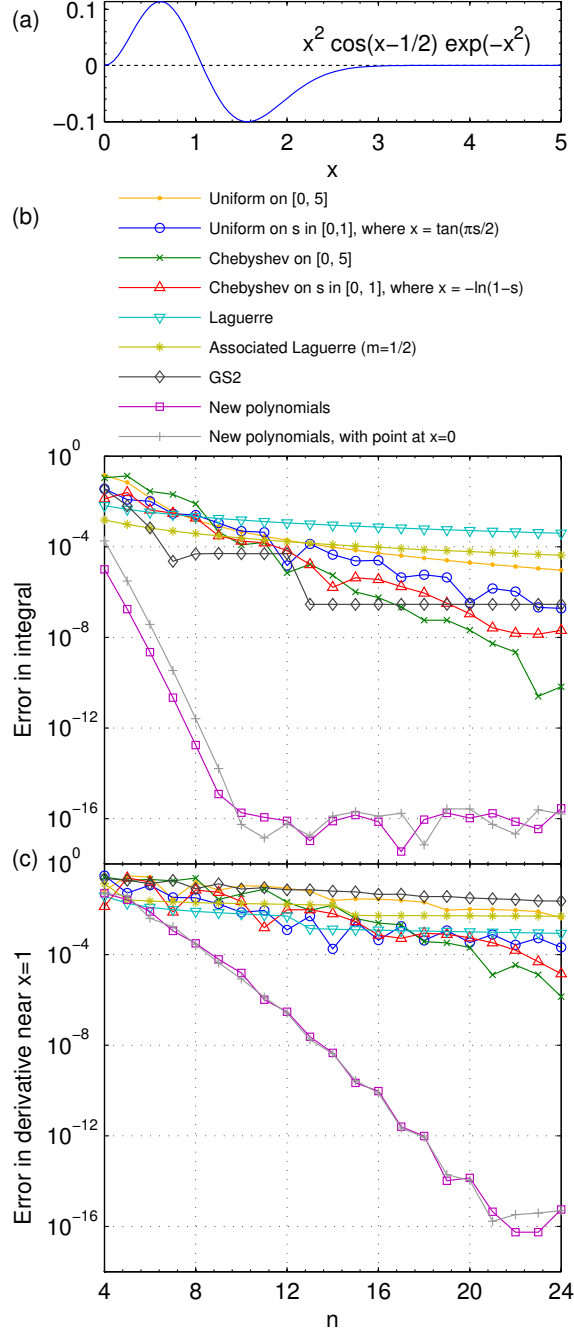


Figure 5: (Color online) A central result of this paper: for the function in (a), typical of the velocity dependence of a distribution function, the new polynomial spectral collocation method described here far outperforms other schemes for both (b) integration and (c) differentiation. In (b)-(c),  $n$  denotes the number of grid points.

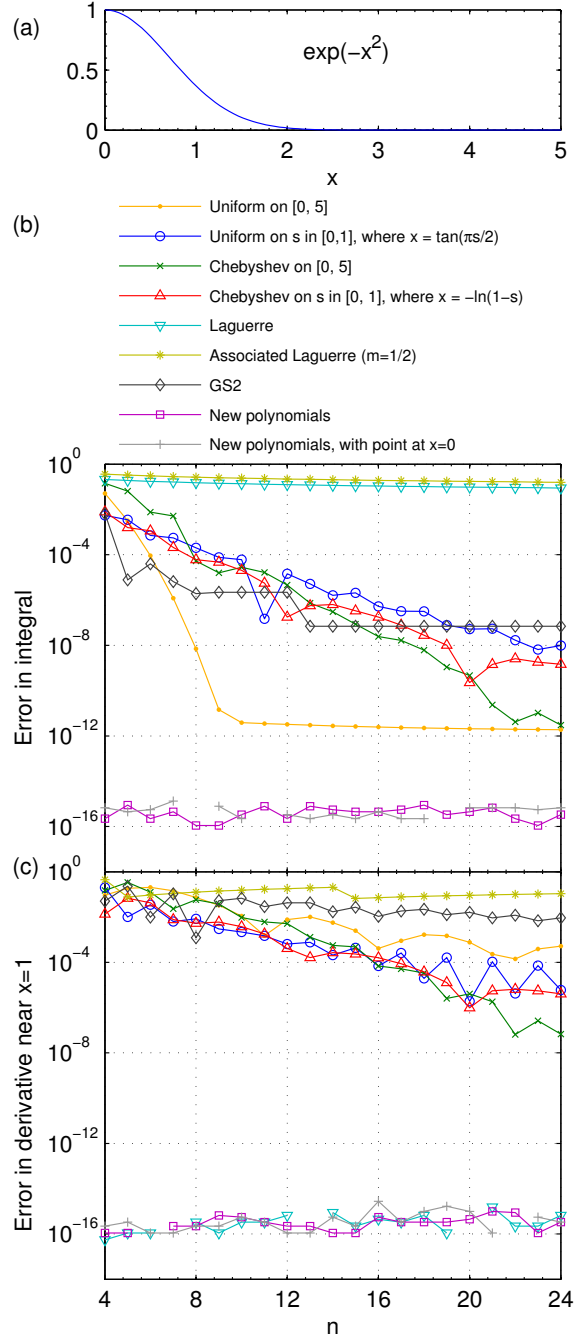


Figure 6: (Color online) Same as Figure 5, but for the function in (a). Missing points indicate zero error to machine precision.

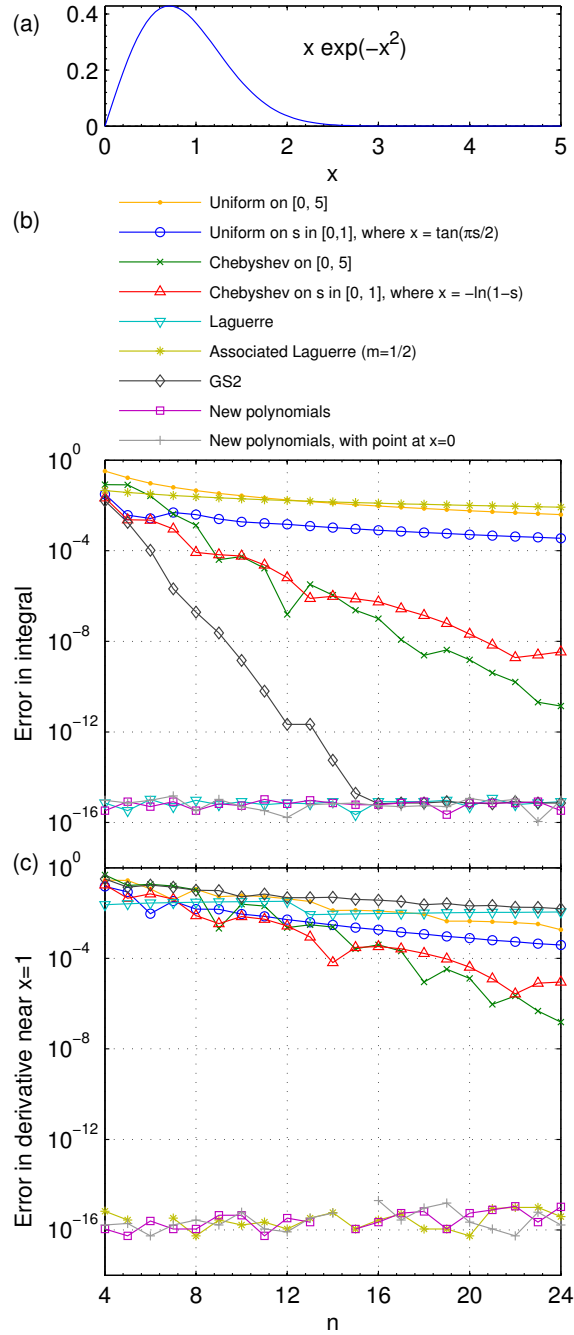


Figure 7: (Color online) Same as Figure 5, but for the function in (a). Missing points indicate zero error to machine precision.

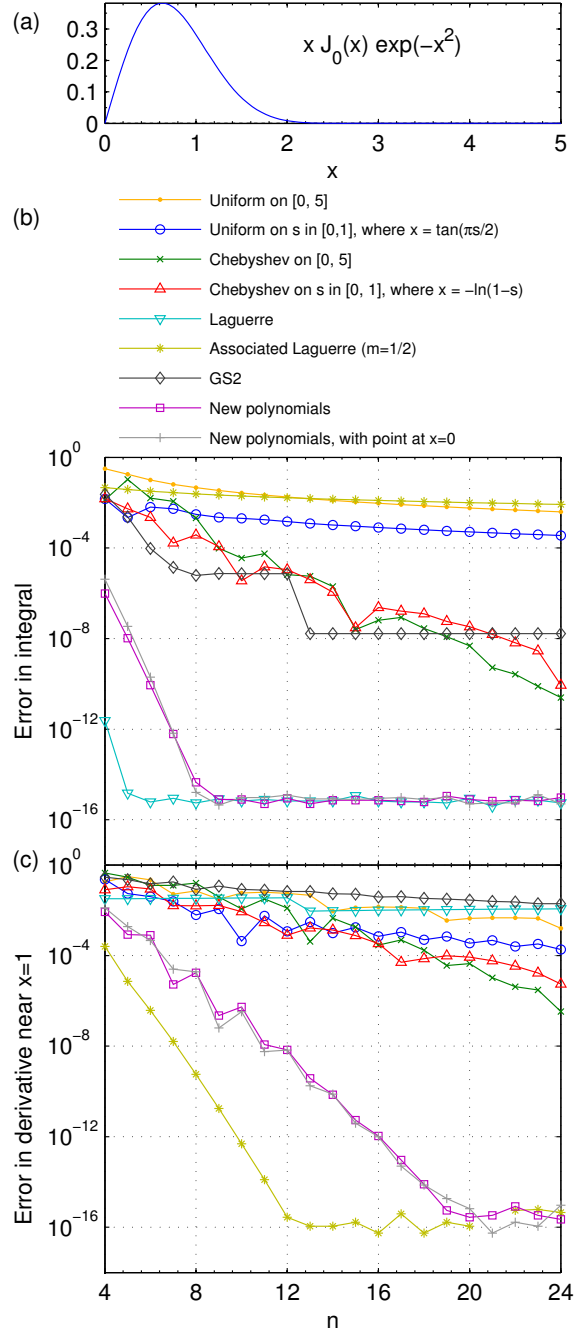


Figure 8: (Color online) Same as Figure 5, but for the function in (a).

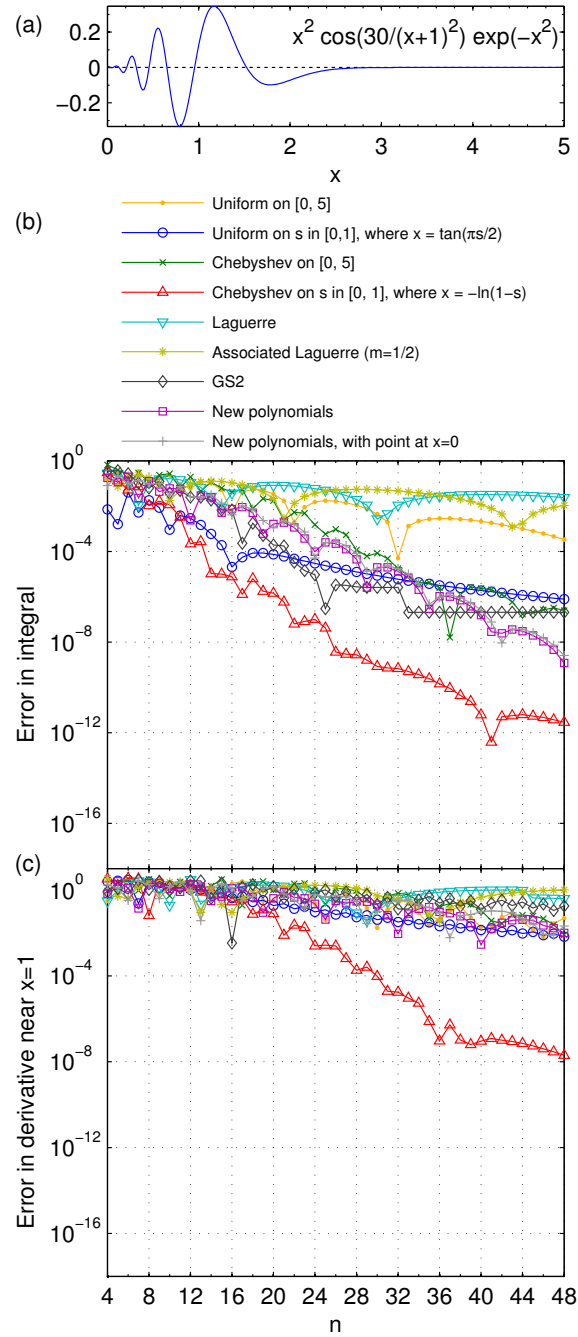


Figure 9: (Color online) Same as Figure 5, but for the function in (a).

Discretization scheme	$n = 4$	$n = 8$	$n = 16$
Uniform on $[0, 5]$	$1 \times 10^{-1}$	$1 \times 10^{-3}$	$5 \times 10^{-5}$
Uniform in $s$ , $x = \tan(\pi s/2)$	$3 \times 10^{-2}$	$2 \times 10^{-3}$	$2 \times 10^{-5}$
Chebyshev on $[0, 5]$	$1 \times 10^{-1}$	$8 \times 10^{-3}$	$6 \times 10^{-7}$
Chebyshev in $s$ , $x = -\ln(1 - s)$	$1 \times 10^{-2}$	$2 \times 10^{-3}$	$4 \times 10^{-6}$
Laguerre	$6 \times 10^{-3}$	$2 \times 10^{-3}$	$7 \times 10^{-4}$
Associated Laguerre ( $m = 1/2$ )	$2 \times 10^{-3}$	$4 \times 10^{-4}$	$9 \times 10^{-5}$
GS2	$3 \times 10^{-2}$	$5 \times 10^{-5}$	$3 \times 10^{-7}$
New polynomials	$1 \times 10^{-5}$	$2 \times 10^{-13}$	$7 \times 10^{-17}$
New polynomials with $x = 0$	$2 \times 10^{-4}$	$3 \times 10^{-12}$	$1 \times 10^{-16}$

Table 1: Error in the integral of  $x^2 \cos(x - 1/2)e^{-x^2}$  for various numbers of grid points  $n$ .

Discretization scheme	$n = 4$	$n = 8$	$n = 16$
Uniform on $[0, 5]$	$6 \times 10^{-2}$	$2 \times 10^{-1}$	$3 \times 10^{-2}$
Uniform in $s$ , $x = \tan(\pi s/2)$	$3 \times 10^{-1}$	$3 \times 10^{-2}$	$4 \times 10^{-4}$
Chebyshev on $[0, 5]$	$3 \times 10^{-1}$	$2 \times 10^{-1}$	$2 \times 10^{-3}$
Chebyshev in $s$ , $x = -\ln(1 - s)$	$1 \times 10^{-2}$	$7 \times 10^{-2}$	$7 \times 10^{-4}$
Laguerre	$4 \times 10^{-2}$	$8 \times 10^{-3}$	$1 \times 10^{-3}$
Associated Laguerre ( $m = 1/2$ )	$1 \times 10^{-1}$	$2 \times 10^{-2}$	$5 \times 10^{-3}$
GS2	$2 \times 10^{-1}$	$8 \times 10^{-2}$	$5 \times 10^{-2}$
New polynomials	$5 \times 10^{-2}$	$3 \times 10^{-4}$	$9 \times 10^{-11}$
New polynomials with $x = 0$	$5 \times 10^{-2}$	$3 \times 10^{-4}$	$8 \times 10^{-11}$

Table 2: Error in the derivative of  $x^2 \cos(x - 1/2)e^{-x^2}$  for various numbers of grid points  $n$ , reported at the grid point closest to  $x = 1$ .



tiation schemes are applied to  $x^a e^{-x^2}$  for various integers  $a$ . The new polynomial scheme gives answers correct to machine precision both for integration (when  $a < 2n$ ) and for differentiation (when  $a < n$ ). The Laguerre or associated Laguerre grids with integer  $m$  can get the derivative correct to machine precision when  $a$  is even, or they can compute the integral correct to machine precision when  $a$  is odd, but they perform poorly in the opposite cases. This behavior follows because the Laguerre method is designed to exactly evaluate  $\int_0^\infty dy y^j e^{-y} = 2 \int_0^\infty dx x^{2j+1} e^{-x^2}$  and to exactly differentiate  $y^j e^{-y}$  for integers  $j$ , but they will perform poorly when the nonanalytic factor  $\sqrt{y}$  is included in the integral or derivative. Conversely, associated Laguerre grids with half-integer  $m$  can compute the integral correct to machine precision when  $a$  is even, or they can compute the derivative correct to machine precision when  $a$  is odd, but they perform rather poorly in the opposite cases. The new polynomials get all cases correct to machine precision for both even and odd  $a$ . In figure (7).b, the GS2 grid can reach machine precision for integration of this special integrand because on the interval  $[2.5, \infty)$ , the function can be exactly integrated by the Laguerre grid, even with just 2 points. Convergence is therefore limited only by convergence of the Legendre grid on the interval  $[0, 2.5]$ .

The integration and differentiation properties of the Laguerre and associated Laguerre grids on  $x^a e^{-x^2}$  for even and odd  $a$  extend in part to all even and odd functions of  $x$ . The latter is illustrated in figure 8, which shows the performance of the various grids for the odd function  $j(x) = x J_0(x) \exp(-x^2)$ , where  $J_0$  is a Bessel function. (We choose  $j(x)$  since functions like it arise in gyrokinetics.) As  $x J_0(x)$  may be expanded in a Taylor series about  $x = 0$  with only odd powers of  $x$ ,  $j(x)$  is integrated accurately on a Laguerre grid, but not differentiated accurately on this grid. Similarly,  $j(x)$  is differentiated accurately on an associated Laguerre grid with half-integer  $m$ , but it is not integrated accurately on this grid. If the  $x$  preceding  $J_0$  is replaced by an even power of  $x$ , the behaviors of integer and half-integer associated Laguerre grids are reversed. However, as shown in figure 8, the new polynomial grids allow very accurate integration and differentiation simultaneously.

We do find that for some functions with small-scale oscillatory structure, the new polynomials may not be the most accurate discretization option. Such a function is illustrated in figure 9. For this function, all grid types show slower convergence of both

the integral and derivative compared to previous functions. (Note the expanded  $n$  axis in figure (9).) The Chebyshev grid in the remapped variable performs best, and the uniform remapped grid also performs well. The superior performance of these grids can be understood in part from their very high density of points near  $x = 0$ , higher even than the new polynomials, helping to resolve the fine structure in this region. For problems in which fine structure is expected in the  $x$  coordinate, such as gyrokinetic turbulence, it may be useful to build codes that can easily switch between several types of grids, since it may not be clear *a priori* which will actually yield faster convergence in practice for the specific problem.

### 3. Rosenbluth potentials

We now consider issues related to the linearized Fokker-Planck collision operator[8], which arises in many kinetic theory problems including neoclassical and turbulence calculations. For many problems of interest, the distribution function  $f_a$  of each species  $a$  is nearly Maxwellian:  $f_a = f_{a0} + f_{a1}$  with  $f_{a1} \ll f_{a0}$  where  $f_{a0} = n_a(\pi^{3/2}v_a^3)^{-1} \exp(-x_a^2)$ ,  $n_a$  is the density,  $v_a = \sqrt{2T_a/m_a}$  is the thermal speed, and  $x_a = v/v_a$ . In this case, the operator  $C_{ab}$  in the kinetic equation of species  $a$  due to collisions with species  $b$  may be linearized about the Maxwellians, and it is given by the sum of the test particle part  $C_{ab}\{f_{a1}, f_{b0}\}$  and the field particle part  $C_{ab}\{f_{a0}, f_{b1}\}$ . The test particle part is[8, 24, 25]

$$C_{ab}\{f_{a1}, f_{b0}\} = \Gamma_{ab}n_b \left[ \frac{\text{erf}(x_b) - \Psi}{v^3} \mathcal{L}\{f_{a1}\} + \frac{1}{v^2} \frac{\partial}{\partial v} \left( 2 \frac{m_a}{m_b} \Psi x_b^2 f_{a1} + \Psi v \frac{\partial f_{a1}}{\partial v} \right) \right] \quad (10)$$

where  $\Gamma_{ab} = 4\pi Z_a^2 Z_b^2 e^4 \ln \Lambda / m_a^2$ ,  $\Psi = [\text{erf}(x_b) - 2\pi^{-1/2} x_b e^{-x_b^2}] / (2x_b^2)$  is the Chandrasekhar function,  $\text{erf}(x_b) = 2\pi^{-1/2} \int_0^{x_b} e^{-w^2} dw$  is the error function,

$$\mathcal{L} = \frac{1}{2} \frac{\partial}{\partial \xi} (1 - \xi^2) \frac{\partial}{\partial \xi} + \frac{1}{2(1 - \xi^2)} \frac{\partial^2}{\partial \varphi^2} \quad (11)$$

is the Lorentz operator,  $\xi = v_{||}/v$ ,  $v_{||} = \mathbf{v} \cdot \mathbf{B}/|\mathbf{B}|$  is the component of the velocity along the magnetic field, and the gyrophase  $\varphi$  is the cylindrical angle in velocity space. The field particle part is

$$C_{ab}\{f_{a0}, f_{b1}\} = \Gamma_{ab}f_{a0} \left[ \frac{2v^2}{v_a^4} \frac{\partial^2 G}{\partial v^2} + \frac{2v}{v_a^2} \left( \frac{m_a}{m_b} - 1 \right) \frac{\partial H}{\partial v} - \frac{2}{v_a^2} H + 4\pi \frac{m_a}{m_b} f_{b1} \right] \quad (12)$$

where  $H$  and  $G$  are the perturbations to the Rosenbluth potentials of species  $b$ , defined by

$$\nabla_{\mathbf{v}}^2 H = -4\pi f_{b1} \quad (13)$$

$$\nabla_{\mathbf{v}}^2 G = 2H \quad (14)$$

with the velocity-space Laplacian  $\nabla_{\mathbf{v}}^2 = v^{-2} [(\partial/\partial v)v^2(\partial/\partial v) + 2\mathcal{L}]$ . (The concise form of the field operator in (12) is derived in Eq. (7) of Ref. [25].) For this paper we focus on drift-kinetic calculations, so we need not retain the gyrokinetic modifications to the test particle[26] and field particle[25] operators that arise when the collision operator is gyroaveraged at fixed guiding center position. We also specialize to the usual case in which only the gyrophase-independent distribution function is considered. (The gyrophase-dependent part of the distribution function is known analytically.)

For the moment, consider a species affected by collisions only with itself (such as the ions in a pure plasma), so we may write the linearized collision operator as  $C_L\{f_{a1}\} = C_{aa}\{f_{a1}, f_{a0}\} + C_{aa}\{f_{a0}, f_{a1}\}$ , and we drop the species subscripts to simplify notation. Suppose also we are solving a linear kinetic equation of the form

$$Df_1 + C_L\{f_1\} = R \quad (15)$$

where  $D$  is some operator describing the kinetic terms and  $R$  is some inhomogeneous term. As we do not know the perturbed Rosenbluth potentials associated with  $f_1$ , we may think of (15) as having the block matrix structure

$$\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & 0 \\ 0 & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} f_1 \\ H \\ G \end{pmatrix} = \begin{pmatrix} R \\ 0 \\ 0 \end{pmatrix}. \quad (16)$$

The first row in this system corresponds to the original kinetic equation (15). The second row corresponds to the Poisson equation (13), and the third row similarly corresponds to (14). The block structure (16) was pointed out previously in Ref. [13]. There is no  $M_{23}$  or  $M_{31}$  element in (16) since  $G$  does not appear in (13) and  $f_1$  does not appear in (14). We may identify  $M_{11}$  with the operator  $D$  in (15), plus the test-particle collision terms, and also the last term in (12) (which does not involve  $H$  or  $G$ .) Also,  $M_{12}$  consists of the  $H$  and  $\partial H/\partial v$  terms in (12),  $M_{13}$  is the  $G$  term in (12),  $M_{21} = 4\pi$ ,  $M_{22} = \nabla_{\mathbf{v}}^2$  with

appropriate boundary conditions for  $H$ ,  $M_{32} = -2$ , and  $M_{33} = \nabla_{\mathbf{v}}^2$  with appropriate boundary conditions for  $G$ .

For several reasons, it is beneficial to discretize the  $\xi$  dependence of  $H$  and  $G$  using Legendre polynomials  $p_\ell(\xi)$ :  $H(z, \xi) = \sum_\ell H_\ell(z) p_\ell(\xi)$  and  $G(z, \xi) = \sum_\ell G_\ell(z) p_\ell(\xi)$  where  $z$  represents any other independent variables. The reasons stem from the fact that the Legendre polynomials are eigenfunctions of  $\nabla_{\mathbf{v}}^2$ :

$$\nabla_{\mathbf{v}}^2 [W(v) p_\ell(\xi)] = p_\ell(\xi) \left[ \frac{1}{v^2} \frac{d}{dv} v^2 \frac{dW}{dv} - \frac{\ell(\ell+1)}{v^2} W(v) \right] \quad (17)$$

for any  $W(v)$ . First, the Legendre representation is extremely efficient, since  $\nabla_{\mathbf{v}}^2 \sim \ell^2$  so  $H_\ell \sim 1/\ell^2$ , and therefore very few modes need to be retained. Second, the Legendre discretization allows efficient treatment of the boundary at large  $v$ , which can be understood as follows. The distribution function will be as small as machine precision for  $x > 6$ , so it is inefficient to locate any grid points beyond this value of  $x$ . However,  $H$  and  $G$  vary not as  $e^{-x^2}$  but rather as powers of  $x$  for large  $x$  (as we will show in a moment), so they are not small on the scale of machine precision even for  $x > 6$ . In fact,  $|G|$  generally increases to infinity with  $x$ , so it is invalid to impose any Dirichlet or Neumann boundary condition on  $G$  at large finite  $x$ . However, using the Legendre expansion above, for  $x > x_{\text{Max}} = 4 - 6$ , the defining equation for  $H$  becomes  $\nabla_{\mathbf{v}}^2 H = 0$ , and so  $H_\ell = A_\ell v^{-(\ell+1)} + B_\ell v^\ell$ . The  $B_\ell \neq 0$  solutions are not physical, and so the Robin boundary condition  $v dH_\ell/dv + (\ell+1)H_\ell = 0$  may be applied at  $x_{\text{Max}}$  to ensure the correct behavior  $H_\ell \propto v^{-(\ell+1)}$ . For the  $G$  potential, the defining equation  $\nabla_{\mathbf{v}}^2 G = 2H$  admits four linearly independent solutions. Two are the homogeneous solutions to  $\nabla_{\mathbf{v}}^2 G = 0$ , and the other two are particular solutions, varying as the homogeneous solutions times  $v^2$ . Thus,  $G_\ell = C_\ell v^{-(\ell+1)} + D_\ell v^\ell + E_\ell v^{1-\ell} + F_\ell v^{\ell+2}$ . As can be seen in Eq. (45) of Ref. [8], the  $D_\ell \neq 0$  and  $F_\ell \neq 0$  solutions are unphysical. To retain both of the remaining two solutions, the boundary condition must be second order, and can be found as follows. Writing  $v^2 d^2 G_\ell/dv^2 + \beta v dG_\ell/dv + \sigma G_\ell = 0$ , the requirements that  $G_\ell \propto v^{-(\ell+1)}$  and  $G_\ell \propto v^{1-\ell}$  be solutions yield two equations for  $(\beta, \sigma)$ . Thus, the condition  $v^2 d^2 G_\ell/dv^2 + (2\ell+1)v dG_\ell/dv + (\ell^2-1)G_\ell = 0$  is the one we impose at  $x_{\text{Max}}$ .

Legendre polynomials tend to be efficient for representing  $f_1$  as well, although for some applications such as bounce-averaged codes, other representations of the pitch-angle dependence of  $f_1$  may be necessary and/or preferable[27, 13, 28]. For simplicity,

we assume for the rest of this discussion that the  $\xi$  dependence of  $f_1$  is discretized in the same Legendre representation as the potentials. If a different discretization is used for the  $\xi$  dependence of  $f_1$ ,  $M_{12}$ ,  $M_{13}$ , and  $M_{21}$  must contain the appropriate matrices for mapping between the representations.

It is much faster to invert the  $M_{22}$  and  $M_{33}$  blocks of (16) than to invert  $M_{11}$ , for two reasons. First, it typically takes many more Legendre modes to represent  $f_1$  than to represent the potentials. Secondly, the kinetic operator  $D$  included in  $M_{11}$  generally involves coupling between real space and velocity space coordinates, whereas  $M_{22}$  and  $M_{33}$  are independent of real space. In fact, since  $M_{22}$  and  $M_{33}$  are diagonal in the Legendre index, the only coordinate with off-diagonal terms in these blocks is  $x$ . Thus, it can be convenient to formally solve the system (16) before discretizing, yielding the form

$$(M_{11} - [M_{12} - M_{13}M_{33}^{-1}M_{32}] M_{22}^{-1}M_{21}) f_1 = R. \quad (18)$$

The fact that  $M_{22}$  and  $M_{33}$  are very sparse and fast to invert means that extremely high resolution in  $x$  can be used for  $H$  and  $G$  for almost no extra cost. This result is fortuitous, because the spectral collocation method described in the previous section cannot be used for  $H$  or  $G$ . This is because, as discussed earlier, the potentials behave as powers of  $x$  rather than as  $\exp(-x^2)$  for large  $x$ . Consequently,  $H$  and  $G$  may be discretized using a very high resolution uniform grid in  $x$  with finite-difference differentiation derivatives used in  $M_{22}$ ,  $M_{33}$ , and  $M_{13}$ . Or, a high resolution Chebyshev grid in  $x$  could be used for the potentials, with spectral differentiation matrices used in  $M_{22}$ ,  $M_{33}$ , and  $M_{13}$ . (For applications in following sections, we use a uniform grid.) In either case, the high resolution grid no longer appears once the matrix multiplications in (18) are performed, so the high resolution does not affect the time to solve the system (18).

A complication introduced by the use of different  $x$  grids for  $f_1$  and the potentials is that functions must be mapped from one grid to the other in  $M_{12}$ ,  $M_{13}$ , and  $M_{21}$ . Mapping from the polynomial grid to the uniform grid can be done with spectral accuracy by evaluating the new polynomials on the uniform grid. An efficient algorithm for this purpose is the *polint* subroutine from Refs. [21, 22, 23]. To map from the uniform grid to the polynomial grid, we may use cubic interpolation. The limited accuracy of this interpolation method is not a problem because the grid resolution for the potentials can

be extremely high at little cost. Or if the potentials are discretized on a Chebyshev grid, the *polint* subroutine may again be used to interpolate with spectral accuracy. To form  $M_{12}$ ,  $M_{13}$ , and  $M_{21}$ , both the “ $f_1 \rightarrow$ potentials” and “potentials  $\rightarrow f_1$ ” interpolations must be expressed as explicit matrices. As a specific example of these concepts, consider that the matrix  $M_{13}$ , which corresponds to the  $G$  term in (12), may be constructed as  $M_{13} = Er\Delta$  where  $E = \Gamma_{ab}f_{a0}(2v^2/v_a^4)$  is a diagonal operator on the  $f_1$  grid,  $r$  is the matrix representing cubic interpolation from the potential grid to  $f_1$  grid, and  $\Delta$  is the finite-difference matrix representation of  $\partial^2/\partial x^2$  on the potential grid.

#### 4. Application 1: Bootstrap current

For a concrete application of all the aforementioned numerical techniques, we consider a calculation of the bootstrap current, a current  $\mathbf{j}$  driven along the magnetic field  $\mathbf{B}$  due to radial gradients of density and temperature in a toroidal plasma. In an axisymmetric (tokamak) plasma consisting of electrons and a single ion species, the bootstrap current has the form[29, 24, 30]

$$\langle \mathbf{j} \cdot \mathbf{B} \rangle = -cIp_e \left[ \mathcal{L}_{31} \frac{1}{p_e} \left( \frac{dp_e}{d\psi} + \frac{dp_i}{d\psi} \right) + \frac{\mathcal{L}_{32}}{T_e} \frac{dT_e}{d\psi} + \frac{\mathcal{L}_{34}\alpha}{ZT_e} \frac{dT_i}{d\psi} \right] \quad (19)$$

using notation consistent with Ref. [30]. In (19),  $c$  is the speed of light,  $I = RB_\zeta$  is the major radius  $R$  times the toroidal magnetic field  $B_\zeta$ ,  $Z$  is the ion charge,  $T_i$  and  $T_e$  are the ion and electron temperatures,  $p_i$  and  $p_e$  are the ion and electron pressures, and  $2\pi\psi$  is the poloidal magnetic flux. The quantity  $\alpha$  is a number obtained by solving the ion kinetic equation, and the quantities  $\mathcal{L}_{31}$ ,  $\mathcal{L}_{32}$ , and  $\mathcal{L}_{34}$  are moments of the electron distribution function for various driving terms, with each distribution obtained by solving the electron kinetic equation. Specifically,

$$\mathcal{L}_{31} = -4\pi^{-1/2} \left\langle \hat{B} \int_{-1}^1 d\xi \int_0^\infty dx x^3 \xi f_{e1}^{(p)} \right\rangle \quad (20)$$

where  $f_{e1}^{(p)}$  is the solution of the normalized electron “drift-kinetic” equation

$$x\xi \frac{\partial f_{e1}^{(p)}}{\partial \theta} - \frac{(1-\xi^2)x}{2\hat{B}} \left( \frac{\partial \hat{B}}{\partial \theta} \right) \frac{\partial f_{e1}^{(p)}}{\partial \xi} - \frac{\nu'}{qR_0\mathbf{b} \cdot \nabla \theta} \hat{C} = \frac{1+\xi^2}{2} x^2 e^{-x^2} \frac{1}{\hat{B}^2} \frac{\partial \hat{B}}{\partial \theta} \quad (21)$$

where  $x = x_e$ , and the independent variables are  $x$ ,  $\xi$ , and the poloidal angle  $\theta$ . Here,  $\mathbf{b} = \mathbf{B}/B$ ,  $B = |\mathbf{B}|$ ,  $\hat{B} = B/B_0$ ,  $B_0 = I/R_0$ ,  $R_0$  is a typical major radius,  $q$  is the safety

factor,  $2\pi\psi$  is the poloidal magnetic flux,  $\nu' = \nu_e q R_0 / v_e$  is a dimensionless measure of collisionality,  $\nu_e = 4\sqrt{2\pi} n_e e^4 \ln \Lambda / (3\sqrt{m_e} T_e^{3/2})$  is the electron collision frequency,  $v_e = \sqrt{2T_e/m_e}$  is the electron thermal speed, and  $\hat{C} = \nu_e^{-1} [C_{ee} + C_{ei}]$ . Angle brackets denote a flux surface average:

$$\langle Y \rangle = (V')^{-1} \int_0^{2\pi} d\theta Y / \mathbf{B} \cdot \nabla \theta \quad (22)$$

for any quantity  $Y$  where

$$V' = \int_0^{2\pi} d\theta / \mathbf{B} \cdot \nabla \theta = \oint d\ell_\theta / B_\theta, \quad (23)$$

$d\ell_\theta$  is the poloidal length element,  $V' = dV/d\psi$ , and  $2\pi V(\psi)$  is the volume enclosed by the flux surface.

By expanding  $C_{ei}$  in  $\sqrt{m_e/m_i} \sim 1/60$  for deuterium, it can be shown that the electron-ion collision operator is dominated[29, 24] by the  $\mathcal{L}$  term in the test particle operator (10). Then since  $\text{erf}(x) - \Psi(x) \rightarrow 1$  for  $x \gg 1$ , the collision operator to use in (21) is  $\hat{C}\{f_{e1}^{(p)}\} = \nu_e^{-1} [C_{ee}\{f_{e1}^{(p)}, f_{e0}\} + C_{ee}\{f_{e0}, f_{e1}^{(p)}\} + \Gamma_{ei} n_i v^{-3} \mathcal{L}\{f_{e1}^{(p)}\}]$ . As the non-Maxwellian ion distribution function  $f_{i1}$  and perturbed ion Rosenbluth potentials are then no longer present in (21), we may solve the single-species electron equation (21) without coupling to the ion equation.

The other coefficients in the bootstrap current are calculated in a similar manner:

$$\mathcal{L}_{32} = -4\pi^{-1/2} \left\langle \hat{B} \int_{-1}^1 d\xi \int_0^\infty dx x^3 \xi f_{e1}^{(T_e)} \right\rangle \quad (24)$$

where  $f_{e1}^{(T_e)}$  is the solution of (21) with an extra factor  $x^2 - 5/2$  multiplying the right-hand side, and

$$\mathcal{L}_{34} = 4\pi^{-1/2} \left\langle \hat{B}^2 \right\rangle^{-1} \left\langle \hat{B} \int_{-1}^1 d\xi \int_0^\infty dx x^3 \xi f_{e1}^{(T_i)} \right\rangle \quad (25)$$

where  $f_{e1}^{(T_i)}$  is the solution of (21) with the right-hand side replaced by  $(3\xi^2 - 1)(x^2/2)e^{-x^2} \partial \hat{B} / \partial \theta$ .

We now demonstrate the solution of (21) and the corresponding equations for  $\mathcal{L}_{32}$  and  $\mathcal{L}_{34}$  using the velocity discretizations described in the preceding sections. Some model must be supplied for the  $\theta$  dependence of the dimensionless quantities  $\hat{B}$  and  $qR_0 \mathbf{b} \cdot \nabla \theta$ . For this example we use the Miller equilibrium model [31] detailed in the appendix.

The  $\theta$  coordinate may be discretized in several ways. In decreasing level of sparsity, some options are finite differencing, a Fourier modal expansion, or Fourier spectral

collocation[7]. (The modal approach need not be fully dense in  $\theta$  since many coupling terms will be exponentially small and can therefore be set to zero.) For results shown here we use the modal approach. The finite-difference and collocation approaches lead to simpler code, since multiplications of the unknown fields by functions of  $\theta$  may then be done pointwise, whereas the modal approach requires matrix multiplication.

If a Gaussian grid is used there is no point at  $x = 0$ , or if the Gauss-Radau grid is used, there is a grid point at  $x = 0$ . In the former case there is no boundary condition imposed at  $x = 0$ , and in the latter case, the boundary condition  $\partial f_{e1}^{(j)}/\partial x = 0$  at  $x = 0$  is imposed for the  $\ell = 0$  Legendre mode. We find results are nearly identical for the two approaches. Results shown below use the grid with no point at  $x = 0$ . No boundary condition needs to be applied  $f_{e1}^{(j)}$  at  $x = \infty$ , since the  $e^{-x^2}$  behavior is automatically enforced by the choice of basis functions.

The reduced system (18) may be solved efficiently using a sparse direct solver, or when the resolution is sufficiently high, faster solution is typically possible using an iterative Krylov-space solver. For the iterative solvers, preconditioning is essential. We find an effective preconditioner can be obtained by eliminating all off-diagonal blocks in the  $x$  coordinate. Calculations shown here are performed using the BICGstab(1) algorithm [32].

Figure 10 shows the convergence of the code outputs  $\mathcal{L}_{31}$ ,  $\mathcal{L}_{32}$ , and  $\mathcal{L}_{34}$  with respect to the various numerical parameters, using the new polynomial grid with no point at  $x = 0$ . For this run we choose the collisionality  $\nu_* = \nu' A^{3/2}$  to be 0.01, where  $A = 3.17$  is the aspect ratio. The excellent convergence with respect to all coordinates is evident. For the scan of  $x_{\text{Max}}$ ,  $Ny$  is varied proportional to  $x_{\text{Max}}$  to maintain constant resolution. The very rapid convergence with the grid size in  $x$  is enabled by the efficient spectral methods described in the preceding sections. Convergence to two digits of precision is plenty for comparison with experiment or for experimental design, and the code is converged beyond this level with only 4-5 grid points in  $x$ . The code execution time is nearly independent of  $Ny$ , the resolution of the uniform  $x$  grid for the potentials, for the reasons discussed in section 3. Fortran/Petsc[33, 34] and Matlab versions of the code run in comparable time, since the rate-limiting step is  $LU$ -factorization of the preconditioner, which is highly optimized in Matlab. Execution times reported in Figure 10 are for the



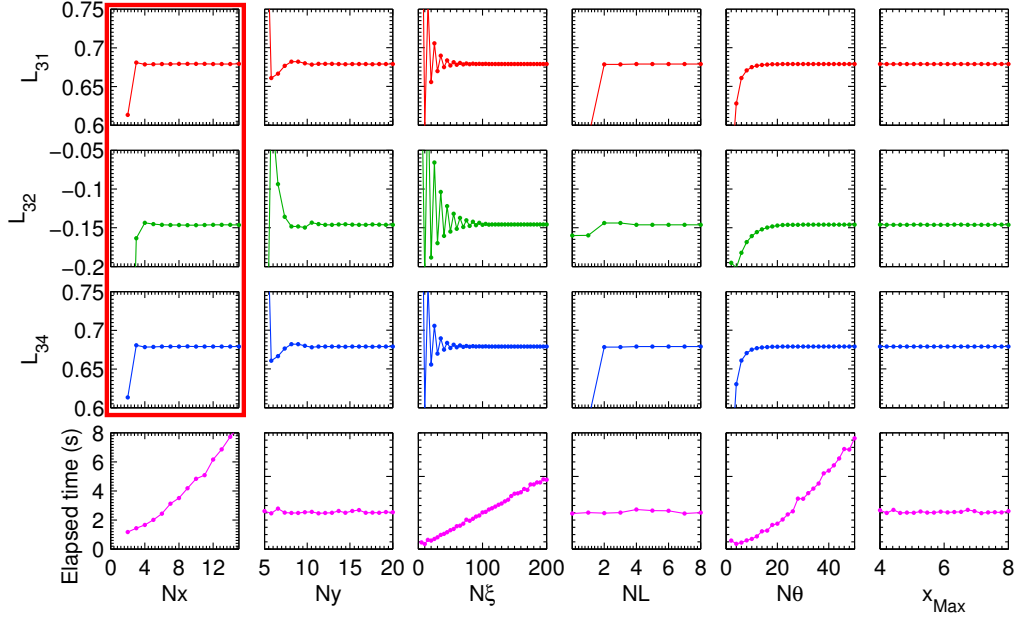


Figure 10: (Color online) Convergence of the bootstrap current coefficients  $\mathcal{L}_{31}$ ,  $\mathcal{L}_{32}$ , and  $\mathcal{L}_{34}$  in (19) from the single-species drift-kinetic code with respect to numerical parameters:  $Nx$  = number of grid points in  $x$  for the polynomial grid,  $Ny$  = number of grid points on the uniform  $x$  grid for the Rosenbluth potentials,  $N\xi$  = number of Legendre modes used to represent the distribution function,  $NL$  = number of Legendre modes used to represent the potentials,  $N\theta$  = number of Fourier modes in  $\theta$ , and  $x_{\text{Max}}$  = cutoff of grid for the potentials. Except for quantities being scanned, the numerical parameters are  $Nx = 6$ ,  $Ny = 20$ ,  $N\xi = 100$ ,  $NL = 4$ ,  $N\theta = 25$ , and  $x_{\text{Max}} = 5$ . The convergence at a very small  $Nx$  enabled by the new discretization scheme described here is a central result of this paper. The collisionality is  $\nu_* = 0.01$ .

Matlab version running on a Dell Precision laptop with Intel Core i7-2860 2.50 GHz CPU and 16 GB memory.

Notice how few Legendre modes ( $\sim 2$ ) need to be retained in the potentials for good convergence, whereas many more Legendre modes ( $\sim 100$ ) are required for the distribution function. The fact that much higher resolution is needed in the  $\xi$  and  $\theta$  coordinates than in the  $x$  coordinate for  $\nu_* \ll 1$  is a result of a boundary layer known to exist in the solution[35], illustrated in Figure 5 of Ref. [36]. This sharp features lies along the curve in the  $(\theta, \xi)$  plane corresponding to the boundary between trapped and circulating particle orbits. The boundary layer widens as collisionality is increased, so for collisionalities higher than the small value used here, many fewer modes in  $\theta$  and  $\xi$  are required, and execution time is then well below 1 second on a laptop.

Further discussion of code benchmarking and results can be found in Ref. [36].

## 5. Multiple species

In computing the bootstrap current, the inter-species coupling was formally negligible, but for other applications it is essential to solve kinetic equations for multiple species simultaneously, retaining cross-species interaction. The thermal speeds of different species may be quite different, so a grid in  $v$  which is appropriate for one species may not be capable of resolving another species. To maintain the high accuracy discussed in Section 2, the distribution function of each species  $f_{a1}$  should be represented on the zeros of the  $P_n(x_a)$  using the normalized speed  $x_a$  for that particular species, as shown in figure 11. Then the field term (12) in the collision operator will require mapping between the  $x_a$  grid and  $x_b$  grid and vice-versa. (The test particle term only involves  $f_{b0}$ , not  $f_{b1}$ , so no remapping of unknown functions is required, and the test particle operator leads to a matrix that is diagonal in species.) Let us consider how to accurately perform these mappings.

The last term in (12), involving  $f_{b1}$  and not the potentials, requires mapping the Maxwellian-like function  $f_{b1}$  from the  $x_b$  grid to the  $x_a$  grid. This corresponds to mapping between grids 1 and 3 in figure 11. This mapping may be accurately done using the *polint* algorithm of Refs. [21, 22, 23], as discussed earlier for mapping from the polynomial grid to the uniform grid. This algorithm works very well both for evaluating values of  $f_{b1}$

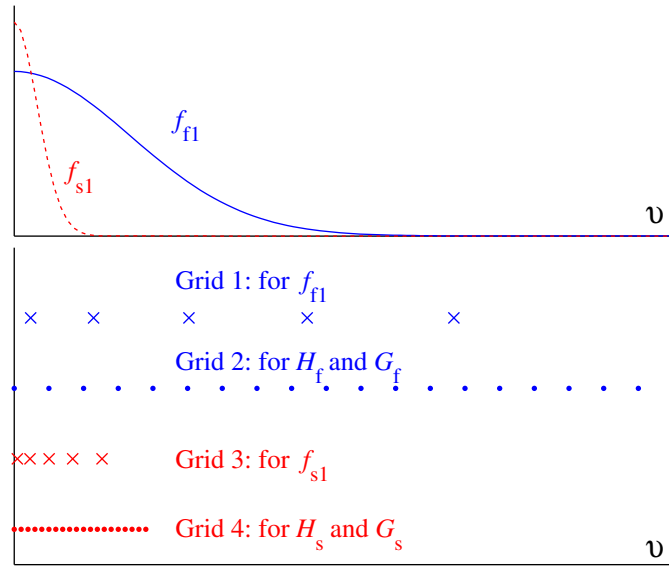


Figure 11: (Color online) For calculations involving  $S$  species,  $2S$  grids in the dimensional speed  $v$  are required: one for each distribution, plus one for the Rosenbluth potentials of each distribution. The polynomial grids for the distributions and the uniform grids for the potentials are each scaled to the thermal speed of the associated species. Here, grids are shown for a slow species (subscript  $s$ ) and a fast species (subscript  $f$ ).

close to  $x_b = 0$  (which is necessary when  $v_b \gg v_a$ , mapping a function known on grid 1 to grid 3) and also for extrapolating to large values of  $x_b$  (which is necessary when  $v_b \ll v_a$ , mapping a function known on grid 3 to grid 1.) Extrapolation is accurate because functional behavior  $\exp(-x_b^2)$  for large  $x_b$  is naturally incorporated into the *polint* algorithm.

Now consider the Rosenbluth potential terms in (12). When  $v_b \gg v_a$ , the potentials must be evaluated close to  $x_b = 0$ . This corresponds to mapping a function known on grid 2 to grid 3 in figure 11. Cubic interpolation is effective for this mapping, just as in the single-species case. However, when  $v_b \ll v_a$ , we must extrapolate the potentials to evaluate them beyond the grid on which they are represented. This corresponds to mapping a function known on grid 4 to grid 1. As discussed in section 3, the  $H$  potential behaves as  $\propto 1/x_b^{\ell+1}$  for  $x_b \rightarrow \infty$ . We find in practice it can be very inaccurate to use the simplistic extrapolation  $H = 0$  for  $x_b > x_{\text{Max}}$ . (In particular, the code described in the following section does not converge well if this extrapolation method is used.) However, the more accurate extrapolation  $H_\ell(x_b) = (x_{\text{Max}}/x_b)^{\ell+1} H_\ell(x_{\text{Max}})$  for  $x_b > x_{\text{Max}}$  works well and leads to good convergence, as we will demonstrate in the next section.

When  $v_b \ll v_a$ , we must also extrapolate  $d^2G/dx_b^2$  to form the first left-hand-side term in (12). Notice from Section 3 that for large  $x_b$ , each Legendre mode  $\ell$  behaves like

$$\frac{d^2G_\ell}{dx_b^2} \approx K_\ell x_b^{-\ell-3} + L_\ell x_b^{-\ell-1} \quad (26)$$

for constants  $K_\ell$  and  $L_\ell$ . It is tempting to make the further approximation  $d^2G_\ell/dx_b^2 \approx L_\ell x_b^{-\ell-1}$ , suggesting the extrapolation  $d^2G_\ell/dx_b^2 = (x_{\text{Max}}/x_b)^{\ell+1} [d^2G_\ell/dx_b^2]_{x_{\text{Max}}}$  for  $x_b > x_{\text{Max}}$ . However, since the  $K_\ell$  term is only algebraically small compared to the  $L_\ell$  term, this extrapolation requires a much higher  $x_{\text{Max}}$  (and therefore many more points in the uniform grid for the potentials) than a more careful extrapolation which accounts for both the  $K_\ell$  and  $L_\ell$  terms. To derive such an extrapolation, we write (26) at the last and penultimate points in the uniform grid for the potentials, which we denote by  $x_{bi}$  and  $x_{bj}$  respectively. Eliminating  $K_\ell$  and  $L_\ell$  by these two equations, we obtain the

improved extrapolation

$$\begin{aligned} \frac{d^2 G_\ell}{dx_b^2} = & \left[ \frac{x_{bi}^{\ell+1}}{(x_{bi}^{-2} - x_{bj}^{-2})x_b^{\ell+3}} + \frac{x_{bi}^{\ell+3}}{(x_{bi}^2 - x_{bj}^2)x_b^{\ell+1}} \right] \left[ \frac{d^2 G_\ell}{dx_b^2} \right]_{x_{bi}} \\ & - \left[ \frac{x_{bj}^{\ell+1}}{(x_{bi}^{-2} - x_{bj}^{-2})x_b^{\ell+3}} + \frac{x_{bj}^{\ell+3}}{(x_{bi}^2 - x_{bj}^2)x_b^{\ell+1}} \right] \left[ \frac{d^2 G_\ell}{dx_b^2} \right]_{x_{bj}}. \end{aligned} \quad (27)$$

We find that this result, which we use for calculations presented below, leads to convergence for much lower  $x_{\text{Max}}$  ( $\sim 4 - 5$ ).

Due to the factor  $f_{a0}$  in (12), every term in (12) will be exponentially small for large  $x_a$ , even if  $v_b \gg v_a$ . Therefore the polynomial  $x_a$  grid is able to accurately resolve the field particle operator when  $v_b \gg v_a$ . However, if  $v_b \ll v_a$ , features in  $f_{b1}$  will be mapped to small  $x_a$ , leading to sharp structures that may not be well resolved by the  $x_a$  grid. For this reason, more grid points in  $x_a$  may be required when the thermal speeds are very different from each other. Fortunately, as shown in Figure 3, the new polynomials have zeros clustered near  $x = 0$ , so grid resolution is quite good in this region, much better than with a Laguerre or associated Laguerre grid. As we will show in the following section, a demonstration for species with very disparate thermal speeds shows that the number of grid points in  $x$  required for good convergence remains quite small.

## 6. Application 2: Impurity flow

As an example application involving multiple species, we consider the neoclassical flows of the main ions and a non-trace impurity species in a tokamak plasma. Due to the low mass of electrons, electrons do not significantly affect the motion of the ions, so we do not need to include electrons in the calculation. The perpendicular flow and the variation of the parallel flow on each flux surface (the so-called Pfirsch-Schlüter flow) can be written explicitly, so it is only the average parallel flow on a flux surface that must be determined from kinetic theory. Following convention, the specific form of the average we compute is  $\langle \mathbf{V} \cdot \mathbf{B} \rangle$ .

For this example calculation, we consider a plasma consisting of deuterium and  $\text{Mo}^{+30}$ . We choose the latter since it is often a plentiful impurity species in metal-wall tokamaks, and its large mass ( $m_{\text{Mo}} = 48m_{\text{D}} = 96 \text{ a.m.u.}$ ) tests the performance of the numerical method when species have disparate thermal speeds. We con-

sider an impurity density  $n_{\text{Mo}}/n_{\text{D}} = 0.0012$ , corresponding to  $Z_{\text{eff}} = 2$ , where  $Z_{\text{eff}} = (n_{\text{D}} + Z_{\text{Mo}}^2 n_{\text{Mo}})/(n_{\text{D}} + Z_{\text{Mo}} n_{\text{Mo}})$ . Since  $Z_{\text{eff}} - 1$  is not  $\ll 1$ , the impurities are not in the trace limit, and so they will have a significant effect on the main-ion flow.

Contributions to the parallel flows arise from radial gradients in the density and temperature of each species, as well as from the radial electric field. For this example, we will compute the contribution from the density gradients, i.e. we formally set  $dT_{\text{D}}/d\psi = 0$ ,  $dT_{\text{Mo}}/d\psi = 0$ , and  $d\Phi/d\psi = 0$ . (Contributions from these other drives could be superposed later due to the linearity of the kinetic equations.) We assume the density scale lengths are equal:  $n_{\text{D}}/(dn_{\text{D}}/d\psi) = n_{\text{Mo}}/(dn_{\text{Mo}}/d\psi)$ . We also take the temperatures of both species to be equal ( $T_{\text{D}} = T_{\text{Mo}} = T$ ).

We normalize both flows with the same factor:

$$\langle \mathbf{V}_{\text{D}} \cdot \mathbf{B} \rangle = U_{\text{D}} \frac{cIT}{en_{\text{D}}} \frac{dn_{\text{D}}}{d\psi} \quad (28)$$

$$\langle \mathbf{V}_{\text{Mo}} \cdot \mathbf{B} \rangle = U_{\text{Mo}} \frac{cIT}{en_{\text{D}}} \frac{dn_{\text{D}}}{d\psi} \quad (29)$$

and so our goal is to compute the unknown dimensionless quantities  $U_{\text{D}}$  and  $U_{\text{Mo}}$ . To do so, we first solve the appropriate normalized drift-kinetic equations

$$\begin{aligned} x_{\text{D}} \xi \frac{\partial f_{\text{D1}}}{\partial \theta} - \frac{(1 - \xi^2)x_{\text{D}}}{2\hat{B}} \left( \frac{\partial \hat{B}}{\partial \theta} \right) \frac{\partial f_{\text{D1}}}{\partial \xi} - \frac{\nu'}{qR_0 \mathbf{b} \cdot \nabla \theta} \frac{C_{\text{D}}}{\nu_{\text{D}}} \\ = \frac{1 + \xi^2}{2} x_{\text{D}}^2 e^{-x_{\text{D}}^2} \frac{1}{\hat{B}^2} \frac{\partial \hat{B}}{\partial \theta} \end{aligned} \quad (30)$$

and

$$\begin{aligned} x_{\text{Mo}} \xi \frac{\partial f_{\text{Mo1}}}{\partial \theta} - \frac{(1 - \xi^2)x_{\text{Mo}}}{2\hat{B}} \left( \frac{\partial \hat{B}}{\partial \theta} \right) \frac{\partial f_{\text{Mo1}}}{\partial \xi} - \frac{\nu'}{qR_0 \mathbf{b} \cdot \nabla \theta} \frac{C_{\text{Mo}}}{\nu_{\text{D}}} \\ = \frac{m_{\text{Mo}}^2 n_{\text{Mo}}}{Z_{\text{Mo}} m_{\text{D}}^2 n_{\text{D}}} \frac{1 + \xi^2}{2} x_{\text{Mo}}^2 e^{-x_{\text{Mo}}^2} \frac{1}{\hat{B}^2} \frac{\partial \hat{B}}{\partial \theta}, \end{aligned} \quad (31)$$

for the distribution functions  $f_{\text{D1}}$  and  $f_{\text{Mo1}}$ , where

$$\begin{aligned} C_{\text{D}} &= C_{\text{DD}}\{f_{\text{D1}}, f_{\text{D0}}\} + C_{\text{DD}}\{f_{\text{D0}}, f_{\text{D1}}\} \\ &\quad + C_{\text{DMo}}\{f_{\text{D1}}, f_{\text{Mo0}}\} + C_{\text{DMo}}\{f_{\text{D0}}, f_{\text{Mo1}}\}, \end{aligned} \quad (32)$$

and

$$\begin{aligned} C_{\text{Mo}} &= C_{\text{MoMo}}\{f_{\text{Mo1}}, f_{\text{Mo0}}\} + C_{\text{MoMo}}\{f_{\text{Mo0}}, f_{\text{Mo1}}\} \\ &\quad + C_{\text{MoD}}\{f_{\text{Mo1}}, f_{\text{D0}}\} + C_{\text{MoD}}\{f_{\text{Mo0}}, f_{\text{D1}}\}. \end{aligned} \quad (33)$$

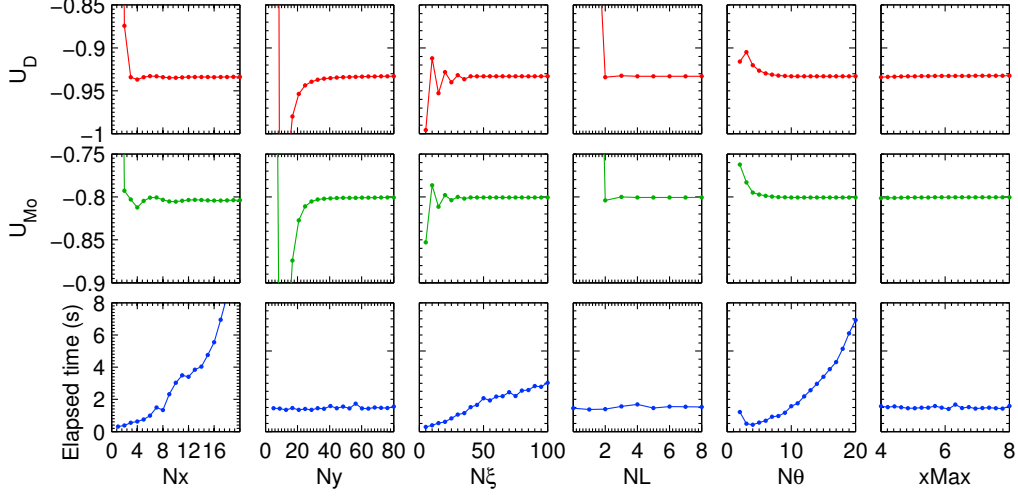


Figure 12: (Color online) Convergence of  $U_D$  and  $U_{Mo}$  – the normalized parallel flows of  $D^{+1}$  and  $Mo^{+30}$  – in the multiple-species drift-kinetic code with respect to numerical parameters defined as in Figure 10. Except for quantities being scanned, the numerical parameters are  $Nx = 8$ ,  $Ny = 80$ ,  $N\xi = 40$ ,  $NL = 4$ ,  $N\theta = 10$ , and  $x_{Max} = 5$ . Good convergence at very small values of  $Nx$  is again apparent. The collisionality is  $\nu_* = 0.01$ .

This time,  $\nu' = \nu_D q R_0 / \nu_D$  in both equations (30)-(31), where  $\nu_D = 4\sqrt{2\pi} n_D e^4 \ln \Lambda / (3\sqrt{m_D} T^{3/2})$ .

Finally, we compute the moments

$$U_D = \frac{4}{\pi^{1/2}} \left\langle \hat{B} \int_0^\infty dx_D x_D^3 \int_{-1}^1 d\xi \xi f_{D1} \right\rangle \quad (34)$$

$$U_{Mo} = \frac{4}{\pi^{1/2}} \left( \frac{m_D}{m_{Mo}} \right)^2 \frac{n_D}{n_{Mo}} \left\langle \hat{B} \int_0^\infty dx_{Mo} x_{Mo}^3 \int_{-1}^1 d\xi \xi f_{Mo1} \right\rangle. \quad (35)$$

We again use the Miller geometry detailed in the appendix. A preconditioning matrix is employed, obtained by dropping not only the blocks that are off-diagonal in  $x$ , as in the single-species code, but also dropping all blocks that are off-diagonal in species.

Figure 12 shows the convergence of the two-species calculation. Calculations are performed for  $\nu_{*D} = \nu' A^{3/2} = 0.01$ . For simplicity, in each run we use the same resolution for both species. As for the single-species case, convergence is achieved at a very small number of grid points in  $x$ , with 1% accuracy achieved by  $Nx = 5$ . Higher  $Ny$  is required, but this causes no decrease in code performance, as shown in the figure, for the reasons discussed in Section 3.

## 7. Discussion

A common feature of kinetic simulations is the need to accurately evaluate both integrals and derivatives of Maxwellian-like functions using a single grid or modal discretization. In this work we have presented a set of methods which allow very accurate integration and differentiation for such functions even with very coarse grids. The central concept is to employ polynomial approximation using the weight  $\exp(-x^2)$  and interval  $[0, \infty)$ , which is equivalent to an expansion in the new orthogonal polynomials (4)-(5). The Stieltjes procedure (detailed in Refs. [15, 16, 17, 18] or Section 4.6.2-3 of Ref. [20], and summarized in Section 2 of the present paper) is used to generate the grid and integration weights. Then the *poldif* and *polint* routines of Refs. [21, 22, 23] are used to generate the differentiation and interpolation matrices. We emphasize that these algorithms from previous references are applicable to any weight function and any integration interval, and were not developed with velocity-space applications in mind. The recognition that these algorithms may be applied to kinetic calculations, using the interval and weight appropriate for the  $v$  or  $v_\perp$  coordinate of Maxwellian-like functions, is novel. Figure 5 is a central result of our work, showing the new method performs far better than other discretization schemes at both integrating and differentiating functions relevant to kinetic theory. The abscissae, integration weights, and differentiation matrices are not significantly more difficult to compute than those for classical orthogonal polynomials, and public-domain source code is available for generating the grid, integration weights, differentiation matrices, and interpolation matrices.

We have also shown how the discretization scheme may be applied when the Fokker-Planck collision operator is included in kinetic calculations. The Rosenbluth potentials in the collision operator cannot be discretized in the same manner as the distribution function because they do not behave as  $\propto \exp(-x^2)$  for large  $x$ . Instead, they may be discretized on a uniform or Chebyshev grid with very high resolution. By the construction (18), this high resolution grid is eliminated through matrix multiplication when the final matrix for the kinetic equation is constructed, and so the grid resolution for the potentials does not affect the size of the final matrix. Consequently, overall code performance and memory requirements are essentially independent of the resolution of the grid for the potentials, and the final matrix size is determined instead by the much coarser resolution



of the grid for the distribution function.

The methods have been demonstrated in several neoclassical calculations for a tokamak plasma, which involve the solution of one or more drift-kinetic equations including collisions. Figure 10 illustrates such a calculation for the bootstrap current, showing that as few as four grid points in  $x$  may be sufficient for convergence of two significant digits (which is more than enough accuracy for comparison to experiment.) When species with disparate masses are included simultaneously, as in the impurity flow computation of figure 12, careful extrapolation of the Rosenbluth potentials must be performed, but as long as this is done, convergence may be achieved for a similar number of grid points in  $x$ .

### Acknowledgements

The authors are grateful to Michael Barnes for assistance regarding the GS2 grid. This work was supported by the Fusion Energy Postdoctoral Research Program administered by the Oak Ridge Institute for Science and Education.

### Appendix A. Miller geometry

Here we present several relations for Miller equilibria, including some useful relations that are not written explicitly in the original reference [31]. Our focus is to compute the quantities needed for the neoclassical calculations described in the main text,  $\mathbf{B} \cdot \nabla \theta$  and the field magnitude  $B$ , both as functions of a poloidal angle  $\theta$ .

The Miller formulation begins with the assumption that the flux surface of interest has the shape

$$R(\theta) = R_0 + r \cos(\theta + x \sin \theta), \quad (\text{A.1})$$

$$Z(\theta) = \kappa r \sin \theta, \quad (\text{A.2})$$

where  $R_0$  is the major radius at the geometric center of the flux surface,  $r$  is the minor radius,  $\kappa$  is the elongation,  $x = \arcsin \delta$ , and  $\delta$  is the triangularity. The aspect ratio  $A$  may be defined by  $A = R_0/r$ . All the quantities  $R_0$ ,  $r$ ,  $\kappa$ ,  $\delta$ , and  $A$  are considered to be functions of the flux surface label  $\psi$ . This  $\theta$  angle is generally not the angle

$\arctan(Z/(R - R_0))$ , nor is it the Boozer or Hamada angle or other straight-field-line coordinate.

Next, consider the standard relation for any  $(q^1, q^2, q^3)$  curvilinear coordinates

$$\nabla q^1 = \frac{1}{J} \frac{\partial \mathbf{x}}{\partial q^2} \times \frac{\partial \mathbf{x}}{\partial q^3} \quad (\text{A.3})$$

where

$$J = \frac{\partial \mathbf{x}}{\partial q^1} \cdot \frac{\partial \mathbf{x}}{\partial q^2} \times \frac{\partial \mathbf{x}}{\partial q^3} = (\nabla q^1 \cdot \nabla q^2 \times \nabla q^3)^{-1} \quad (\text{A.4})$$

is the Jacobian, and here we choose  $(q^1, q^2, q^3) = (\psi, \zeta, \theta)$  with  $\zeta$  the toroidal angle. Recalling that any axisymmetric field may be written  $\mathbf{B} = \nabla \zeta \times \nabla \psi + I \nabla \zeta$ , the quantity  $\mathbf{B} \cdot \nabla \theta$  we are seeking is equal to  $1/J$ . To evaluate the derivatives in (A.3)-(A.4), we write the position vector as

$$\mathbf{x} = \mathbf{e}_X R \cos \zeta + \mathbf{e}_Y R \sin \zeta + \mathbf{e}_Z Z, \quad (\text{A.5})$$

where  $(\mathbf{e}_X, \mathbf{e}_Y, \mathbf{e}_Z)$  are the Cartesian unit vectors. Differentiating (A.5), we obtain, after some algebra,

$$J = \frac{1}{\mathbf{B} \cdot \nabla \theta} = \frac{Y}{\partial \psi / \partial r} \quad (\text{A.6})$$

where

$$\begin{aligned} Y = & \kappa r R \left\{ \sin(\theta + x \sin \theta) [1 + x \cos \theta] (1 + s_\kappa) \sin \theta \right. \\ & \left. + \cos \theta \left[ \frac{\partial R_0}{\partial r} + \cos(\theta + x \sin \theta) - s_\delta \sin(\theta + x \sin \theta) \sin \theta \right] \right\}, \end{aligned} \quad (\text{A.7})$$

$s_\kappa = (r/\kappa) \partial \kappa / \partial r$  and  $s_\delta = (1 - \delta^2)^{-1/2} r \partial \delta / \partial r$ . Then from the magnitude of (A.3),

$$|\nabla \psi| = \frac{rR}{J} \sqrt{\{\sin(\theta + x \sin \theta) [1 + x \cos \theta]\}^2 + (\kappa \cos \theta)^2}. \quad (\text{A.8})$$

Defining  $B_0 = I/R_0$ , the field magnitude may be written

$$B = \sqrt{B_\theta^2 + (B_0 R_0 / R)^2} \quad (\text{A.9})$$

where  $B_\theta = |\nabla \psi|/R = (\text{A.8})/R$  is the poloidal field. The  $B_\theta$  calculated in this manner from (A.6)-(A.8) is precisely Eq. (37) of Ref. [31].

Equations (A.6)-(A.9) with (A.1) are the desired expressions for the two quantities we need,  $B$  and  $\mathbf{B} \cdot \nabla \theta$ , albeit in terms of the quantity  $\partial \psi / \partial r$  in (A.6). It is usually

preferable to specify the safety factor  $q$  in place of  $\partial\psi/\partial r$ . To do so, we observe

$$q = \frac{1}{2\pi} \int_0^{2\pi} d\theta \frac{d\zeta}{d\theta} = \frac{1}{2\pi} \int_0^{2\pi} d\theta \frac{\mathbf{B} \cdot \nabla \zeta}{\mathbf{B} \cdot \nabla \theta}. \quad (\text{A.10})$$

Then using (A.6),

$$\frac{\partial\psi}{\partial r} = \frac{B_0 R_0}{2\pi q} \int_0^{2\pi} d\theta \frac{Y}{R^2} \quad (\text{A.11})$$

where the integral may be evaluated numerically.

In summary, the magnetic geometry is fully specified by the following seven dimensionless parameters:  $A$  (aspect ratio),  $\kappa$  (elongation),  $\delta$  (triangularity),  $q$  (safety factor),  $s_\kappa$  (elongation shear),  $s_\delta$  (triangularity shear), and  $\partial R_0/\partial r$  (related to the Shafranov shift). Since only the normalized quantities  $\hat{B} = B/B_0$  and  $qR_0 \mathbf{b} \cdot \nabla \theta$  appear in the code, there is no need to specify the two dimensional parameters  $r$  (minor radius) and  $B_0$  (magnetic field at the geometric center of the flux surface.) The additional dimensionless parameters  $s$  (global magnetic shear) and  $\alpha$  (normalized pressure gradient) are discussed in Ref. [31] but are not needed for the calculations here.

The calculations for figures 10 and 12 were performed using the following parameters, identical to Ref. [31]:  $A = 3.17$ ,  $\kappa = 1.66$ ,  $\delta = 0.416$ ,  $s_\kappa = 0.70$ ,  $s_\delta = 1.37$ ,  $\partial R_0/\partial r = -0.354$ , and  $q = 3.03$ .

## References

- [1] F. Valentini, P. Veltri, A. Mangeney, J. Comp. Phys. 210 (2005) 730–751.
- [2] M. Barnes, W. Dorland, T. Tatsuno, Phys. Plasmas 17 (2010) 032106.
- [3] R. Numata, G. G. Howes, T. Tatsuno, M. Barnes, W. Dorland, J. Comp. Phys. 229 (2010) 9347–9372.
- [4] A. Pataki, L. Greengard, J. Comp. Phys. 230 (2011) 7840–7852.
- [5] E. A. Belli, J. Candy, Plasma Phys. Controlled Fusion 54 (2012) 015015.
- [6] C. W. Clenshaw, A. R. Curtis, Numerische Mathematik 2 (1960) 197–205.
- [7] L. N. Trefethen, Spectral methods in Matlab, SIAM, Philadelphia, 2000.
- [8] M. N. Rosenbluth, W. M. MacDonald, D. L. Judd, Phys. Rev. 107 (1957) 1–6.
- [9] L. Pareschi, G. Russo, G. Toscani, J. Comp. Phys. 165 (2000) 216–236.
- [10] Z. Xiong, R. H. Cohen, T. D. Rognlien, X. Q. Xu, J. Comp. Phys. 227 (2008) 7192–7205.
- [11] J. A. Spencer, J.-Y. Ji, E. D. Held, J. Comp. Phys. 231 (2012) 6192–6206.
- [12] S. K. Wong, V. S. Chan, Plasma Phys. Controlled Fusion 53 (2011) 095005.
- [13] B. C. Lyons, S. C. Jardin, J. J. Ramos, Phys. Plasmas 19 (2012) 082515.

- [14] M. Barnes, I. G. Abel, W. Dorland, D. R. Ernst, G. W. Hammett, P. Ricci, B. N. Rogers, A. A. Schekochihin, T. Tatsuno, *Phys. Plasmas* 16 (2009) 072107.
- [15] W. Gautschi, *Math. Comp.* 22 (1968) 251–270.
- [16] W. Gautschi, *SIAM J. Sci. Stat. Comp.* 2 (1982) 289–317.
- [17] W. Gautschi, *ACM Trans. Math. Software* 20 (1994) 21–62.
- [18] W. Gautschi, ORTHPOL, Accessed October 6, 2012. <http://www.netlib.org/toms/726>.
- [19] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Webnote No. 3, Rev. 1: Implementation of Stiel, Accessed October 6, 2012. <http://www.nr.com/webnotes?3>.
- [20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes*, 3rd. ed., Cambridge University Press, 2007.
- [21] J. A. C. Weideman, S. C. Reddy, *ACM Trans. Math. Software* 26 (2000) 465–519.
- [22] J. A. C. Weideman, S. C. Reddy, A Matlab Differentiation Matrix Suite, Accessed October 6, 2012. <http://dip.sun.ac.za/~weideman/research/differ.html>.
- [23] J. A. C. Weideman, S. C. Reddy, DMSUITE: Matlab Differentiation Matrix Suite, Accessed October 6, 2012. <http://www.mathworks.com/matlabcentral/fileexchange/29>.
- [24] P. Helander, D. J. Sigmar, *Collisional Transport in Magnetized Plasmas*, Cambridge University Press, Cambridge, 2002.
- [25] B. Li, D. R. Ernst, *Phys. Rev. Lett.* 106 (2011) 195002.
- [26] P. J. Catto, K. T. Tsang, *Phys. Fluids* 20 (1977) 396–401.
- [27] J. G. Cordey, *Nucl. Fusion* 16 (1976) 499–507.
- [28] J. B. Parker, P. J. Catto, *Plasma Phys. Controlled Fusion* 54 (2012) 085011.
- [29] F. L. Hinton, R. D. Hazeltine, *Rev. Mod. Phys.* 48 (1976) 239–308.
- [30] O. Sauter, C. Angioni, Y. R. Lin-Liu, *Phys. Plasmas* 6 (1999) 2834–2839.
- [31] R. L. Miller, M. S. Chu, J. M. Greene, Y. R. Lin-Liu, R. E. Waltz, *Phys. Plasmas* 5 (1998) 973–978.
- [32] G. L. G. Sleijpen, D. R. Fokkema, *Electr. Trans. Num. Anal.* 1 (1993) 11–32.
- [33] S. Balay, J. Brown, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Web page, Accessed October 6, 2012. <http://www.mcs.anl.gov/petsc>.
- [34] S. Balay, J. Brown, , K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, H. Zhang, PETSc Users Manual, Technical Report ANL-95/11 - Revision 3.3, Argonne National Laboratory, 2012.
- [35] F. L. Hinton, M. N. Rosenbluth, *Phys. Fluids* 16 (1973) 836–854.
- [36] M. Landreman, D. R. Ernst, *Plasma Phys. Controlled Fusion* 54 (2012) 115006.